

MaDDash Quick Install Guide v4

06/2021

All commands here are based on a default, minimal CentOS 7 installation. Other Linux variants may require modifications to these commands to work properly (yum vs apt-get, repository locations, etc). All commands require root access, so either use *sudo* before each command or use *sudo su* and save yourself some typing. Commands as entered will be in *italic print*.

MaDDash Initial Install

Install perfSONAR/MaDDash using perfSONAR-centralmanagement package

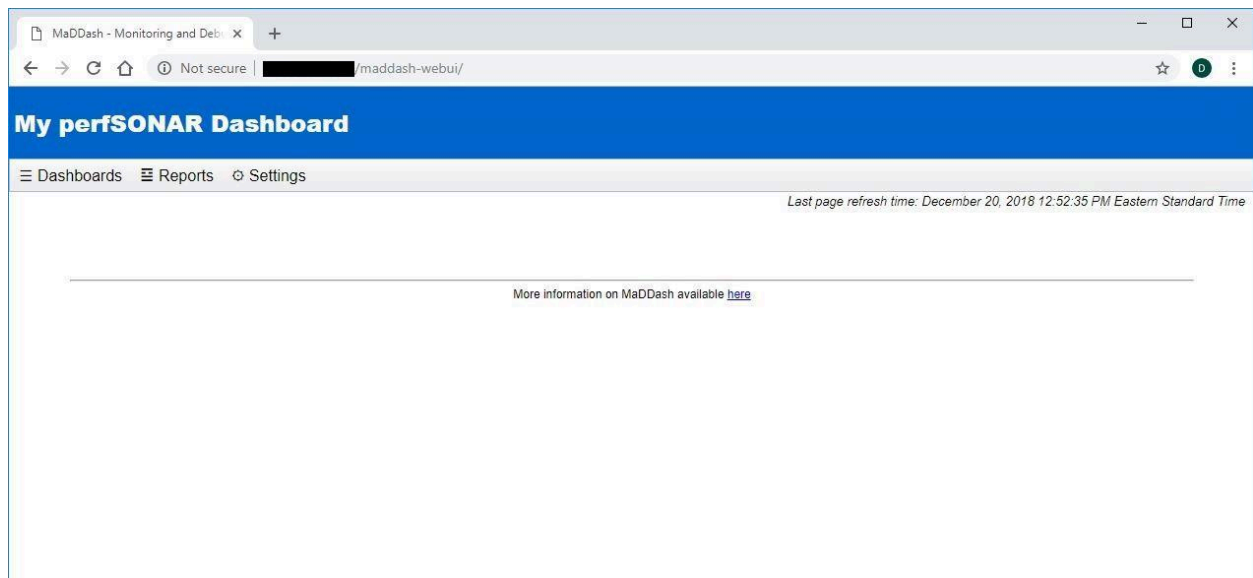
- *yum update* (Always do this first!)
- *yum install epel-release*
- *yum install*
http://software.internet2.edu/rpms/el7/x86_64/latest/packages/perfSONAR-repo-0.10-1.noarch.rpm
 - If it doesn't work, check here to make sure you have the current URL:
https://docs.perfsonar.net/maddash_install.html
- *yum clean all*
- *yum install perfsonar-centralmanagement*
 - **OPTIONAL:** *yum install perfsonar-psconfig-maddash* (installs MaDDash without Measurement Archive)

Configure Firewall - Open Port 80 & 443

- *firewall-cmd --permanent --add-port=80/tcp*
- *firewall-cmd --permanent --add-port=443/tcp*
 - (*--permanent* flag causes the rule to be added permanently, otherwise it will disappear next firewall reload/system reboot)
- *firewall-cmd --reload*

MaDDash Configuration

At this point you should have a working, unconfigured MaDDash instance. You can test this by going to “http(s)://(IP or hostname)/maddash-webui”. You should get a web page that looks like this:



Now it's time to build a configuration file and publish it on the local and remote machines. Config files and templates are located in

`/usr/share/doc/perfsonar/psconfig/`

Examples in this section use vi as the editor, but feel free to use any editor of your choice (vim/nano/pico/emacs/etc). This guide isn't intended to be a comprehensive resource for building these files, but that information can be found at

http://docs.perfsonar.net/psconfig_maddash_agent.html#configuring-templates

- `vi /usr/share/doc/perfsonar/psconfig/skeleton.json` (or ANY of the provided template files) and save it as a new file when you are done. Here's a few tips:
 - Names entered under “address” & “group” are internal identifiers only
 - Tests, schedules, groups, and tasks all need to be defined
 - Run `jq . name-of-file.json` to check for errors. If it simply echoes back your script there are no syntax errors in the JSON file. Otherwise it will give you the line and column number where the error occurred
- `psconfig publish /path/to/name-of-file.json`
 - Publishes the file after it's complete and moves it to the web

- *esmond_manage add_user_ip_address **example-user ip-address***
 - Allows pS nodes defined in the config to archive their data
 - **example-user** is any username you want to put here and isn't referenced anywhere in the config file. It can be thought of as a way to group nodes together in esmond under a common name.
 - **ip-address** is the address of the remote pS node as defined in the config file. Must enter this command for each pS node, or use CIDR notation to enter a subnet (e.g. 10.0.0.1/24)
 - **To verify addresses already added:** *sudo esmond_manage list_user_ip_address*
 - **To remove addresses already added:** *delete_user_ip_address ip-address*
- *psconfig remote add "https://address-of-archive/psconfig/name-of-file.json"*
 - Lets the archive know where the JSON config file lives
- Restart services
 - *systemctl enable cassandra*
 - *systemctl restart cassandra*
 - *systemctl restart maddash-server*
 - *systemctl restart psconfig-maddash-agent*

A quick http reload option for the MaDDash:
https://ip-address-of-archive/maddash-webui/etc/config.json

pS Node Configuration

There's a couple of commands you need to run on each node that will allow them to consume the remote configuration file and join the MaDDash.

- *psconfig remote add --configure-archives "https://address-of-archive/psconfig/name-of-file.json"*
 - Note the "--configure-archives" flag. This tells the node to store its results remotely on the measurement archive.
- Restart services
 - *systemctl restart psconfig-pscheduler-agent*

At this point you should have a working MaDDash. Below are some basic troubleshooting tips if things aren't going as planned.

Basic Troubleshooting

Most issues with MaDDash stem from a problem with the JSON config file. Double check everything there first and make sure that names, IPs, variables, etc are all correct.

BE PATIENT! The MaDDash won't populate instantly. Results displayed on the MaDDash are the average of the last 3 tests. If, for example, bandwidth testing is happening once every 4 hours, it'll be a while before results start showing up.

On the Measurement Archive:

- *less /var/log/maddash/psconfig-maddash-agent.log*
 - Check the MaDDash log for errors if nothing is displayed on the dashboard.
- *psconfig remote list*
 - See if the archive location configuration is correct.

On the Nodes:

- *curl -k https://ip-address-of-archive/psconfig/name-of-file.json*
 - Assuming all is working well, you can grab the config file from the archive using this command on any configured node. Useful for making sure that the node has access to the Measurement Archive and that the configuration file has been published successfully.
- *less /var/log/perfsonar/psconfig-pscheduler-agent.log*
 - Check the pscheduler-agent log for errors if tasks don't seem to be running.
- *psconfig remote list*
 - See if the archive location configuration is correct. Make sure that "configure-archives" : true is present in the output.
- *psconfig pscheduler-stats*
 - See if there are any scheduled tasks from the remote archive
- *pscheduler monitor*
 - See if pscheduler tasks are running and completing on schedule

MaDDash YAML Edits (Optional)

Remove default ESnet display:

- *vi /etc/maddash/maddash-webui/config.json*
 - change defaultDashboard to NAMEOFDASHBOARD

Auto redirect to correct MaDDash. This causes a specific dashboard to load automatically when a user goes to “https://your-domain-name” instead of the normal blank page where a user has to select a dashboard.

- *vi /etc/httpd/conf.d/apache-maddash-root-redirect.conf* (Creates new Apache config file) and add these lines:
 - RewriteEngine On
 - RedirectMatch ^/\$ /maddash-webui/(rest of the string to your specific MaDDash instance).
 - For example, if your specific MaDDash instance is located at “http://your-domain-name/maddash-webui/index.cgi?dashboard=YOUR%DASH”, then this line would read “RedirectMatch ^/\$ /maddash-webui/index.cgi?dashboard=YOUR%DASH”
- Restart services
 - *systemctl restart httpd*

Advanced Mesh Setting Resources

http://docs.perfsonar.net/psconfig_maddash_agent.html

More specifically (and especially important) -

http://docs.perfsonar.net/psconfig_maddash_agent.html#adjusting-thresholds-and-other-parameters